



Рис. 8.

$$d = \frac{1}{(1+r)^B}$$
, де B – відношення днів до кількості днів у році.

Інтерполювання ставки дисконтування простіше, оскільки графік цієї функції має форму, близьку до прямої (див. рис. 8).

Проте розрахунок процентної ставки зі ставки дисконтування може призвести до виникнення досить значних похибок обчислень, а тому він не бажаний.

1. Zhuoshi Liu. Moorad Choudhry. Yield curve fitting, 2005.
2. Baki Isa. Yield curve estimation by spline-based models, 2006.

3. Sederberg T. W. Computer Aided Geometric Design Course Notes, 2007.

K. Degtiarova, V. Popov, A. Trygub

YIELD CURVE FITTING

Are examined the contemporary methods of approximations of one of the indices of bank activity – by profitableness curve. The use of cubic spline-functions in the task of the approximation of these curves is based. Numerical calculations and analysis of the obtained results are represented.

УДК 681.3

Гломозда Д. К.

КООРДИНАЦІЯ ВЗАЄМОДІЇ КОРИСТУВАЧІВ У КОЛАБОРАТИВНІЙ СИСТЕМІ ДИСТАНЦІЙНОЇ ОСВІТИ ВНЗ

В роботі описано прототип системи, яка пов'язує автоматизовану систему управління навчальним закладом (АСУНЗ) з системою керування навчанням (СКН). Демонструються принципи практичного застосування технології рівневого контролю (floor control) для координації дій користувачів двох систем.

Вступ

Стаття продовжує цикл робіт [1, 2], присвячених побудові моделі програмної системи підтримки дистанційної взаємодії в масштабах Інтернет (далі – ПСПДВІ), що ґрунтується на технології рівневого контролю [3, 4]. Принцип рівневого контролю полягає в наданні користувачам доступу до спільних ресурсів на основі визначеної для кожної групи користувачів політики обслуговування. В попередніх роботах була запропонована модель ПСПДВІ мовою мереж Петрі та описано багатоагентну систему, яка

моделювала роботу складових такої системи. Тут одержані результати використовуються для побудови прототипу координаційної системи, що пов'язує автоматизовану систему управління навчальним закладом (далі – АСУНЗ) з системою керування навчанням (далі – СКН) з метою узгодження дій користувачів двох систем й уникнення конфліктних ситуацій.

Використані технології

До складу колаборативної системи дистанційної освіти для вищого навчального закладу

входять АСУНЗ, СКН та система забезпечення координації роботи перших двох складових.

За СКН було взято використовуване в НаУКМА середовище Moodle, розгорнуте в середовищі WampServer 2.0 [5]. В якості АСУНЗ – систему MAMS [6]. Це веб-застосування, програмна частина якого розгортається на сервері Apache Tomcat 5.0.25, а база даних – на MS SQL Server 8.0.

Третя складова – наша власна EPCS (Education Process Coordination System – система координації навчального процесу), яка буде координувати роботу користувачів Moodle та MAMS, запобігаючи виникненню конфліктних ситуацій. Прикладом такої конфліктної ситуації може бути випадок, коли викладач виставляє студентам оцінки за підсумковий тест з курсу, а методист кафедри запитує підсумкову іспитово-залікову відомість для цього курсу. Якщо їхні дії не узгоджувати, методист кафедри може отримати відомість із заниженими оцінками, що призведе до неприємностей.

Для полегшення інтеграції з системою Moodle за мову програмування для написання EPCS було обрано PHP, а за платформу розгортання – той самий WampServer 2.0, до складу якого, крім HTTP-сервера Apache, входять СКБД MySQL з програмою phpMyAdmin – веб-застосуванням для адміністрування баз даних MySQL, та інтерпретатор мови PHP з великим набором розширень, зокрема `php_mysql` та `php_mssql` для роботи з базами даних під управлінням СКБД MySQL та Microsoft SQL Server.

Опис системи

Програмна частина системи EPCS – це веб-застосування, написане мовою PHP з використанням JavaScript, яке складається з трьох модулів: модуль входу в систему та реєстрації нових користувачів, модуль викладача та модуль методиста кафедри.

Ще однією важливою складовою системи є база даних «epcs», яка розміщується на сервері БД, наданим WampServer. Вона складається з двох таблиць, «users» (користувачі) та «active_tasks» (активні задачі). В таблиці «users» зберігаються дані про зареєстрованих користувачів: логіни, паролі, тип користувача (адміністратор, методист чи викладач), імена в системах MAMS та Moodle, а також прапорець, який показує, чи перебуває даний користувач зараз у системі. В таблиці «active_tasks» зберігаються відомості про задачі, які в даний момент виконуються в системі, а саме тип задачі, ім'я користувача-виконавця та ідентифікаційний номер задіяного при цьому ресурсу.

Крім цього, до складу системи входить файл `data_config.inc`, в якому у форматі PHP-змінних зберігаються дані, необхідні для під'єднання до

баз даних систем MAMS, Moodle та власної бази даних системи, а саме: адреса сервера БД, логін і пароль користувача, від чийого імені ми працюватимемо з БД, та назва потрібної нам БД. Потім цей файл додається до програм, що його використовують, за допомогою функції `require(«data_config.inc»)`.

Модуль входу та реєстрації нових користувачів включає в себе такі програмні файли:

- а) *index.html* – стартова сторінка системи з полями, куди зареєстровані користувачі вводять свої імена та паролі. Система не передбачає гостьового доступу до своїх функцій, тож нові користувачі мають зареєструватися, натиснувши на кнопку «New user»;
- б) *newuser.php* – програма, що запускається, коли користувач натискає кнопку «New user». Вона пропонує користувачу ввести ім'я, під яким він хоче бачити себе в системі; свою роль (методист чи викладач); імена, під якими він зареєстрований в системах Moodle та MAMS, а також бажаний пароль (з підтвердженням). Після заповнення всіх полів слід натиснути на кнопку «Register», що запускає програму *newuser_check.php*;
- в) *newuser_check.php* – програма перевірки правильності введених користувачем при реєстрації параметрів. Спершу перевіряється ідентичність двох введених екземплярів паролів. Потім за допомогою SQL-запитів до баз даних Moodle та MAMS перевіряється, чи зареєстровані такі користувачі в цих системах (а в разі, якщо користувач хоче зареєструватися як викладач, чи є він викладачем хоча б одного курсу в Moodle), і чи є вже користувач з таким іменем в EPCS. Крім того для запобігання створенню дублюючих облікових записів в базі даних EPCS міститься інформація про те, які імена користувачів Moodle та MAMS вже були використані, і якщо новий користувач назвався вже зайнятим іменем, система його про це повідомить і попросить ввести інше ім'я. Якщо ж усі перевірки пройшли вдало, система повідомляє користувачу, що він може входити і працювати в системі;
- г) *login.php* – програма обробки введених користувачами імен та паролів. За допомогою SQL-запиту програма звертається до бази даних «epcs» і перевіряє, чи існує такий користувач і чи збігається його пароль із уведеним. Якщо всі перевірки проходять вдало, програма запитує з бази даних роль користувача в системі та завантажує відповідний модуль. Водночас ім'я користувача передається через рядок браузера, щоб у подальшому його можна було використовувати в запитах до баз даних.

Модуль викладача включає засоби, якими вчителі послуговуються в своїй повсякденній роботі. До його складу входять файли:

а) *teacher.php* – основна робоча сторінка вчителя, на якій містяться його робочі інструменти, зокрема засіб для виставлення студентам оцінок за підсумковий тест. Містить два випадючих списки для вибору курсу та групи, причому перелік курсів обмежений лише тими, які даний користувач саме викладає. Для того, щоб список груп формувався залежно від того, який курс обрано, використовується вставка мовою JavaScript, що перевантажує сторінку кожного разу, коли змінюється обраний курс. Її текст:

```
<script language=JavaScript>
function reload(form, user_id){
var course_id =
form.course_id.options[form.course_id.options.
selectedIndex].value;
var group_id=form.group_id.options[form.group_id.
options.selectedIndex].value;
self.location='teacher.php?username=' + user_id
+ '&course_id=' + course_id + '&group_id='
+ group_id;
}
</script>
```

З випадючими списками функція reload зв'язується таким чином:

```
echo "<select name='course_id' onchange='\"
reload(this.form, ' $username')\"><option
value='\">Select course...</option>\"";
```

Слід зазначити, що при цьому випадючі списки повертаються в початковий стан, тому щоб обраний користувачем залишався активним елементом списку курсів, його ідентифікаційний номер передається через рядок браузера, потрапляючи завдяки цьому в асоціативний масив \$_GET – локальну змінну середовища, інформація з якої може бути вільно прочитана будь-якою PHP-програмою, що міститься на даній сторінці. Відповідний фрагмент коду виглядає так:

```
@ $course_id=$_GET['course_id'];
...
while($row_courses = mysql_fetch_array($result_
courses))
{
if($row_courses['id']==@$course_id)
{
echo "<option selected value = '$row_courses[id]'>
$row_courses[fullname]</option>\". "<BR>\";
} else {
echo "<option value = '$row_courses[id]'> $row_
courses[fullname]</option>\";
}
}
}
```

Після того, як викладач обрав курс і групу, програма надсилає до бази даних Moodle такий запит:

```
SELECT DISTINCT mdl_user.id, mdl_user.last-
name, mdl_user.firstname, mdl_grade_grades.
rawgrademin, mdl_grade_grades.rawgrademax
FROM (mdl_user INNER JOIN mdl_grade_grades
ON mdl_user.id = mdl_grade_grades.userid)
INNER JOIN mdl_grade_items ON mdl_grade_
grades.itemid = mdl_grade_items.id
WHERE mdl_grade_items.courseid = $course_id
AND mdl_user.id IN (SELECT userid FROM mdl_
groups_members WHERE groupid = $group_
id)
AND mdl_grade_grades.itemid IN (SELECT id
FROM mdl_grade_items WHERE mdl_grade_
items.itemname = 'Final Test')
ORDER BY mdl_user.lastname";
```

За результатами запиту будується форма введення оцінок з відповідною кількістю текстових полів вводу і кнопка «Enter grades», після натискання на яку введена інформація передається в програму *enter_grades.php*. Разом з нею через приховані (hidden) поля передається службова інформація, а саме: ім'я викладача, кількість студентів в даній групі, ідентифікатори курсу та групи, мінімальна та максимальна оцінки за підсумковий тест з даного курсу, а також ідентифікатор задачі внесення оцінок, під яким вона зареєстрована в таблиці «active_tasks» бази даних «epcs»;

б) *enter_grades.php* – програма обробки та запису введених викладачем оцінок до бази даних. Вона зчитує надіслану попередньою формою інформацію з асоціативного масиву \$_REQUEST і перевіряє її на відповідність таким умовам: а) вона має бути числом; б) вона має бути не меншою за мінімально можливу оцінку і не більшою за максимально можливу оцінку.

Якщо ці умови виконуються, програма фіксує оцінку до бази даних та відповідним чином переобраховує сумарну кількість набраних студентом балів. Відповідні запити мають такий вигляд:

```
UPDATE mdl_grade_grades
SET finalgrade = $grade_to_enter
WHERE userid = $id_grade_to_enter
AND itemid in (SELECT id FROM mdl_grade_items
WHERE courseid = $course_id AND itemname
= 'Final Test')
```

```
UPDATE mdl_grade_grades
SET finalgrade = '$sum[Sum]'
WHERE userid = $id_grade_to_enter
AND itemid in (SELECT id FROM mdl_grade_items
WHERE courseid = $course_id AND itemtype =
'course')
```

Тут \$id_grade_to_enter – ідентифікаційний номер студента, якому виставляється оцінка, \$grade_to_enter – ця сама оцінка, а \$sum[Sum] – сума оцінок, які даний студент одержав протягом курсу. Обраховується вона так:

```
SELECT SUM(finalgrade) AS 'Sum'
FROM mdl_grade_grades
WHERE userid = $id_grade_to_enter
AND itemid in (SELECT id FROM mdl_grade_items
WHERE courseid = $course_id AND itemtype
!= 'course')
```

За результатами роботи програми користувачу надається звіт, з якого він може дізнатися, чи були оцінки виставлені і чи були якісь помилки з уведеними даними або з самою базою даних.

Модуль методиста кафедри включає інструменти, якими методисти послуговуються в повсякденній роботі. До його складу входять файли:

- а) *dep_methodist.php* – як і аналогічний йому *teacher.php*, містить робочі інструменти методиста кафедри. Наразі вони представлені засобом складання заліково-іспитових відомостей для потрібних курсу та групи, котрі, як і на сторінці викладача, обираються за допомогою двох випадючих списків, наповнення для яких береться з бази даних середовища Moodle. Коли потрібна група обрана, з'являється кнопка «Generate», що запускає програму генерації відомості *gen_grades.php*;

- б) *gen_grades.php* – програма генерації заліково-іспитових відомостей. Робиться це за допомогою такого запиту:

```
SELECT DISTINCT mdl_user.lastname, mdl_user.
firstname, mdl_grade_grades.finalgrade
FROM (mdl_user INNER JOIN mdl_grade_grades
ON mdl_user.id = mdl_grade_grades.userid)
INNER JOIN mdl_grade_items ON mdl_grade_
grades.itemid = mdl_grade_items.id
WHERE mdl_grade_items.itemtype='course'
AND mdl_grade_items.courseid = $course_id
AND mdl_user.id IN (SELECT userid FROM mdl_
groups_members WHERE groupid = $group_
id)
ORDER BY mdl_user.lastname
```

Отримані результати виводяться на екран у вигляді таблиці.

Координація дій користувачів, в даному випадку викладача і методиста, відбувається через вже згадану таблицю «active_tasks» бази даних «epcs». Кожна задача, що потребує використання потенційно конфліктного ресурсу, реєструється в цій таблиці за допомогою відповідного SQL-запиту. Наприклад, реєстрація

процесу виставлення оцінок групі студентів виглядатиме так:

```
INSERT
INTO active_tasks
VALUES ($i, 'grades_entering', '$username',
$group_id),
```

де \$i – збільшене на одиницю максимальне з наявних у таблиці значень ідентифікаційного номера задачі, який є первинним ключем і не може повторюватися (або 1, якщо таблиця порожня), «grades_entering» (введення оцінок) – тип задачі, \$username – ім'я користувача (викладача), який виставляє оцінки, \$group_id – ідентифікатор групи. Тепер користувачі, що намагатимуться здійснити яку-небудь операцію з оцінками цієї групи, натомість отримуватимуть повідомлення про те, що в даний момент певний викладач якраз вводить оцінки для цієї групи, і слід зачекати. Якщо ж якомусь користувачу просто знадобиться вивести на екран склад цієї групи, він зробить це без перешкод, оскільки ця дія не стосується оцінок.

Після того, як оцінки були введені і внесені до бази даних, запис про цю задачу видаляється з бази даних «epcs» за допомогою такого запиту:

```
DELETE FROM active_tasks
WHERE id = '$task_id'
```

де \$task_id – ідентифікатор задачі внесення оцінок, переданий з програми *teacher.php* через приховане поле HTML-форми. Аналогічний запит, але вже такий, що розповсюджується на всі задачі, зареєстровані даним користувачем, запускається програмою *logout.php*, що викликається, коли користувач виходить з системи.

Висновки

Як засвідчили випробування, система EPCS ефективно перешкоджає виникненню позаштатних ситуацій, пов'язаних з неузгодженим доступом користувачів до даних, що підтверджує адекватність запропонованих раніше моделей. Наразі ведеться робота з інтегрування EPCS до MAMS та Moodle так, щоб користувачу достатньо було увійти в одну з цих систем, яка брала б на себе всю майбутню роботу, за потреби звертаючись до іншої через координаційну систему. В перспективі планується впровадження такої системи в усьому НаУКМА.

1. Гломозда Д. К. Формальна модель функціонування колаборативного середовища / Д. К. Гломозда, М. М. Глибовець // Тези доповідей Третьої Міжнародної конференції «Теоретичні та прикладні аспекти побудови програмних систем» (ТААПСД'2006). – Київ (Україна), 2006. – С. 225–230.
2. Гломозда Д. К. Використання програмних агентів для організації ефективного управління роботою програмної системи підтримки мережної співпраці / Д. К. Гломозда //

Наукові записки. Комп'ютерні науки. – 2008. – Т. 86. – С. 53–56.

3. Dommel H.-P. Floor control for multimedia conferencing and collaboration / H.-P. Dommel, J. J. Garcia-Luna-Aceves // Multimedia Systems. – Springer-Verlag, 1997. – Vol. 5. – P. 23–38.
4. Dommel H.-P. Networking Foundations for Collaborative Computing at Internet Scope / H.-P. Dommel, J. J. Garcia-Luna-Aceves // International ICSC Congress on Intelligent Systems

- and Applications, Symposium on Interactive and Collaborative Computing (ICC'2000). – Wollongong (Australia), 2000.
5. Wampserver 2.0 – <http://www.wampserver.com/>
6. Глибовець М. М. Розробка системи управління навчально-го закладу на прикладі НаУКМА / М. М. Глибовець,

С. А. Іващенко, О. О. Крусъ // Наукові праці: науково методичний журнал. Серія «Комп'ютерні науки». – Т. 57, Вип. 44. – Миколаїв: Вид-во МДГУ ім. Петра Могили, 2006. – С. 214–219.

D. Glomozda

COORDINATION OF USERS INTERACTION IN COLLABORATIVE DISTANT EDUCATION SYSTEM FOR HIGHER EDUCATIONAL INSTITUTION

A prototype of a system connecting an automated educational institution management system with a learning management system is described. Principles of practical implementation of floor control technology to coordinate these systems' users' actions are demonstrated.

УДК 519.8

Бірюков Д. С., Заславський В. А., Євгєнко В. В., Франчук О. В.

МОДЕЛЮВАННЯ ТА ОЦІНКА СЦЕНАРІЇВ ЗАГРОЗ ДЛЯ ОБ'ЄКТІВ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

В роботі запропоновано математичну модель сценаріїв загроз для об'єктів критичної інфраструктури та алгоритми оцінки ймовірності здійснення загроз.

Сталий розвиток всіх сфер життєдіяльності держави і суспільства безпосередньо залежить від безпеки об'єктів критичної інфраструктури (ОКРІС), вразливість яких протягом останнього десятиріччя стрімко зростає [1]. Причиною є, з одного боку, об'єктивні тенденції розвитку сучасних складних технічних систем (збільшення масштабу, зростання структурної складності, зміна організаційних та виробничих факторів), в таких науковомісних галузях, як енергетика, обчислювальна техніка та телекомунікації, а з іншого – посилення соціальної та економічної залежності від якості, надійності та безпеки функціонування ОКРІС. Також негативно впливає на ситуацію економічна криза, яка спричинила гострий дефіцит фінансових ресурсів. В ряді публікацій за останні роки, наприклад [2, 3], підкреслюється необхідність розвитку методології розв'язання задач аналізу та синтезу безпечних ОКРІС.

У зв'язку з активізацією тероризму в різних сферах діяльності виникає необхідність перегляду заходів та розробки нових підходів до забез-

печення системної безпеки, створення фізичного захисту об'єктів критичної інфраструктури, впровадження нових методів контр-тероризму [4–8].

Міжнародний досвід у використанні проектної загрози для захисту об'єктів критичної інфраструктури ґрунтується головним чином на досвіді захисту ядерних матеріалів і ядерних установок та відображений у документах МАГАТЕ, а також державних установ [9–11]. Загальні підходи, сформовані для ядерної галузі, можуть також бути застосовані до інших об'єктів, які потребують високого рівня впевненості у тому, що їх захист є ефективним. Таким чином, високу практичну важливість має створення методів і засобів формалізації та структуризації процесів розробки, оцінки проектної загрози на основі ймовірнісних методів для подальшого формування фізичного захисту об'єктів критичної інфраструктури.

Метою роботи є представлення математичних засобів опису сценаріїв загрози та ймовірнісної оцінки реалізації загроз.